

Perpustakaan SKTM

SISTEM IMEJ KEPADA DATA

**PROJEK ILMIAH TAHAP AKHIR II
WXES3182**

NAMA : MUHAMAD NAJIB BIN ZAKARIA

NO.MATRIK : WEK000284

PENYELIA : PUAN NORJIHAN BTE ABDUL GHANI

MODERATOR : PUAN ABRIZAH

TARIKH : 18 FEBRUARI 2003

Abstrak

Sistem ini boleh terima imej dari format yang berbeza-beza seperti TIFF, BMP, GIF, dan JPEG. Format imej yang juga boleh diterima tetapi mesti mempunyai format asas piksel. Saiz imej adalah bebas, tiada kekangan untuk keperluan saiz imej dalam sistem. Begitu juga dengan saiz sebarang objek yang wujud dalam imej.

Jenis data yang digunakan untuk mewakili data objek adalah jenis data binari. Data binari lebih mudah digunakan dalam proses didalam sistem dan bersesuaian dengan kegunaannya dipelbagai pihak dan sistem masa kini. Proses menukarkan imej kepada data tidak rumit dan mudah dikendalikan oleh pengguna.

Objek di dalam imej berbentuk pelbagai jenis, segitiga, segiempat, bulat dan lain-lain. Objek akan dikesan dengan menggunakan teknik pengecaman objek melalui pendekatan pengesanan penjuru. Data akan diambil daripada objek tersebut dan dijadikan data objek. Data objek ini akan dipaparkan kepada pengguna pada skrin sistem.

Projek ini dibangunkan dengan perancangan yang rapi dan hasil penyelidikan dijalankan untuk mendapatkan maklumat dan analisa projek. Pelbagai kaedah dan teknik digunakan sepanjang tempoh pembangunan projek bermula sejak dari penerimaan projek sehingga penyerahan projek. Segala masalah yang timbul dihadapi dengan baik dan pemilihan jalan penyelesaian terbaik adalah satu cara penyelesaiannya.

Penghargaan

Setinggi penghargaan ditujukan khas kepada penyelia, Puan Norjihan Bte Abdul Ghani, pensyarah Fakulti Sains Komputer & Teknologi Maklumat atas nasihat dan tunjuk ajar beliau, dimana telah memberikan saya panduan yang amat berguna.

Ribuan terima kasih juga diucapkan kepada ahli keluarga, rakan-rakan dan moderator saya iaitu Puan Norizan Binti Mohd Yasin, yang banyak membantu dalam menyumbangkan buah fikiran dan pendapat bagi perancangan projek ini. Sumbangan sentiasa berkongsi pendapat dan masalah sepanjang tempoh perancangan projek dari semua pihak terlibat amat saya hargai.

Dengan sokongan yang diberikan kepada saya, insyaAllah projek ini akan berjalan lancar dan mencapai kejayaan. Semoga segala pengetahuan dan pengalaman dalam menyiapkan projek ini akan dapat diaplikasikan pada masa hadapan dan memberi faedah kepada semua yang terlibat.

Diharap sistem ini akan memberi faedah kepada pihak yang memerlukannya dan kejayaan sistem ini berfungsi dengan baik dalam keadaan persekitaran yang sebenar amat bermakna bagi saya. Di masa-masa hadapan kelak diharap dapat saya tempuhi dengan yang terbaik dalam bidang lapangan projek yang mencabar lagi menguji diri saya jika diberi peluang lagi.

Senarai Kandungan

Senarai Tajuk	Halaman
Abstrak	i
Penghargaan	ii
Senarai Jadual	v
Senarai Rajah	vi
1.0 PENGENALAN	
1.1 Definisi Sistem	1
1.2 Objektif Sistem	2
1.3 Skop Projek	2
1.4 Penjadualan Projek	3
2.0 KAJIAN LITERASI	
2.1 Definisi Sistem	7
2.2 Kajian Terhadap Sistem Yang Telah Wujud	8
2.3 Persekitaran Teknologi Pengaturcaraan	12
2.4 Persekitaran Sistem Pengendalian	13
2.5 Kajian Teknik Pengecaman Objek Yang Digunakan	15
3.0 METODOLOGI	
3.1 Pengenalan	18
3.2 Model Air Terjun (Waterfall Model)	19
4.0 ANALISA SISTEM	
4.1 Keperluan Fungsian	25
4.2 Keperluan Bukan Fungsian	25
4.3 Pertimbangan-pertimbangan Lain	

4.3.1 Keperluan Perisian	26
4.3.2 Keperluan Perkakasan	26
5.0 REKEBENTUK SISTEM	
5.1 Carta Aliran Data	27
5.2 Rekebentuk Senibina	27
5.3 Rekebentuk Antaramuka	28
6.0 PELAKSANAAN SISTEM	30
6.1 Pengkodan Sistem	30
6.1.1 Struktur Kawalan	30
6.1.2 Algoritma	31
6.1.3 Struktur Data	31
6.1.4 Alat Pembangunan	31
6.1.5 Alatan Grafik	33
6.1.5.1 Adobe Photoshop 7.0	33
6.2 Dokumentasi Program	34
6.2.1 Dokumentasi Dalaman	34
6.2.2 Dokumentasi Luaran	34
6.3 Ringkasan	35
7.0 PENGUJIAN DAN PENYELENGGARAAN	35
7.1 Pengenalan	35
7.2 Jenis-jenis Pengujian	36
7.2.1 Pengujian Unit	36
7.2.2 Pengujian Modul	36
7.2.3 Pengujian Integrasi	37
7.2.4 Pengujian Sistem	38

7.2.5 Pengujian Pengguna	38
7.3 Perancangan Ujian	39
7.4 Fasa Penyelenggaraan	39
7.5 Ringkasan	39
8.0 PENILAIAN SISTEM	40
8.1 Pengenalan	40
8.2 Kekuatan Sistem	40
8.2.1 Mesra Pengguna	40
8.2.2 Skrin yang Menarik	40
8.3 Had Sistem	41
8.3.1 Masalah Data	41
8.3.2 Penggunaan Tetikus	41
8.4 Peningkatan Masa Hadapan	41
8.5 Ringkasan	41
9.0 MASALAH, PENYELESAIAN DAN KESIMPULAN	42
9.1 Masalah-masalah	42
9.1.1 Pengetahuan	42
9.1.2 Paparan Maklumat	43
9.2 Kesimpulan	44
SENARAI RUJUKAN	45
LAMPIRAN A	
LAMPIRAN B	

Senarai Jadual

Senarai Tajuk	Halaman
Jadual 1.1: Jadual Perancangan Projek	6
Jadual 2.1 : Jujukan nombor	15
Jadual 5.1 : Ciri-ciri antaramuka	29

Senarai Rajah

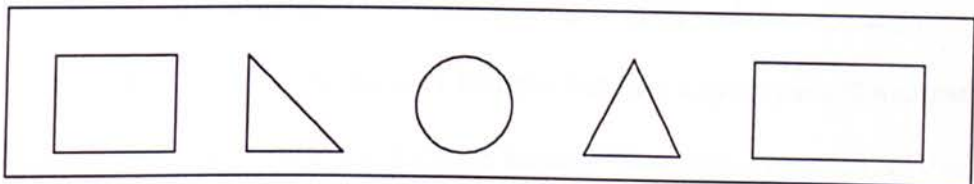
Senarai Tajuk	Halaman
Gambarajah 1.1: Contoh bentuk-bentuk objek	1
Gambarajah 2.1 : Contoh imej	7
Gambarajah 2.2 : Arah Kompas	17
Gambarajah 3.1 : Model Air Terjun	19
Gambarajah 5.1 : Carta Aliran Data	27
Gambarajah 5.2 : Senibina sistem imej kepada data	28

1.0 PENGENALAN

1.1 Definisi Sistem

Pertukaran imej kepada data adalah menukarkan grafik kepada bentuk data. Grafik imej mempunyai beberapa format yang berlainan seperti bit-mapped (BMP), tagged image file format (TIFF), graphics interchanged format (GIF) dan joint photographic expert groups (JPEG). Jenis data yang digunakan adalah data binari yang mempunyai nilai 0 dan 1.

Imej dalam pelbagai format ini akan dijalankan proses pengecaman objek untuk mengesan objek di dalam imej tersebut. Objek yang sudah dikesan akan diambil bacaan data binarinya sebagai data objek. Objek disini bermaksud bentuk corak paparan didalam imej seperti bentuk bulat, empat segi tepat, empat segi sama, dan segitiga (Sila rujuk gambarajah 1.1 : contoh bentuk-bentuk objek).



Gambarajah 1.1: Contoh bentuk-bentuk objek

Proses pengecaman imej dijalankan bagi mengenal pasti objek-objek yang hadir di dalam imej tersebut. Konsep yang digunakan adalah melabelkan peristiwa spatial primitif yang dialami oleh piksel. Piksel adalah unit asas imej dimana ia mewakili unit diskrit bagi koordinat spatial 2 dimensi.

Data objek yang diperolehi akan dipaparkan terus ke antaramuka sistem untuk paparan pengguna. Pengguna akan mengetahui data kepada objek di dalam imej tersebut.

1.2 Objektif Sistem

Tujuan sistem ini dibangunkan adalah untuk dapatkan data objek di dalam imej grafik. Imej akan dikecam untuk mengesan sebarang objek yang wujud dan kemudian lokasi objek dicatitkan. Data pada lokasi itu akan diambil sebagai data objek. Contohnya objek bulat dikesan di dalam imej, lokasi bulat dicatit dan data binari pada lokasi piksel bulat diambilkkan.

Pelbagai bentuk objek boleh dikesan dan ditukarkan kepada data. Data-data ini boleh dijadikan sumber kaedah analisis data dalam pelbagai bentuk dan cara dengan ketelitian teknikal yang diperlukan oleh pihak pengguna yang mahukan bantuan kedalaman teknikal imej. Perwakilan data ini amat berguna kepada pengaturcaraan untuk membangunkan algoritma untuk proses struktur data yang memerlukan kitar manipulasi data imej.

1.3 Skop Projek

Skop projek menerangkan garis sempadan dan had dalam membangunkan projek ini. Sistem ini boleh dijanakan sendiri tanpa perlu bahagian lain. Ini bermaksud sistem tidak memerlukan sumber dari perisian lain, fail perkongsian atau sistem server untuk berfungsi. Sistem ini tidak bergantung kepada mana-mana program lain kerana

mempunyai fail dan datanya sendiri yang diperlukan. Fail dan data telah dihasilkan semasa pembangunan sistem.

Jangkaan projek ini akan disiapkan dan berjaya berfungsi sepenuhnya adalah pada awal bulan Mac 2003. Bagi jangkaan tempoh kerja dijalankan adalah selama lapan bulan bermula pada 15 Julai 2002 sehingga akhir bulan Februari 2003. Perancangan kos pembangunan projek ini diteliti dan konsep penjimatan di ambil beratkan memandangkan projek ini membangunkan perisian yang tidak besar. Perisian pembangunan yang digunakan adalah yang biasa dan mudah diperolehi. Kos sampingan seperti tenaga manusia, kos elektrik dan lain-lain tidak membebankan keseluruhan kos.

Pembangunan projek ini hanya memerlukan sebuah komputer peribadi untuk analisis dan pembangunan sistem. Perisian bahasa pengaturcaraan juga dimuatkan ke dalam komputer sebagai 'workstation'. Perisian grafik akan digunakan juga untuk membekal alat grafik kepada pembangunan sistem.

1.4 Penjadualan Projek

Pembangunan sistem imej kepada data ini melibatkan beberapa langkah utama iaitu pembentukan idea dan merealisasikan idea tersebut secara praktikal. Analisa terperinci tentang sistem yang akan dibangunkan juga merupakan langkah yang penting selain mengumpul keperluan-keperluan pengguna.

Metodologi pembangunan sistem ini menggunakan Modul Air Terjun atau Kitar Hayat Pembangunan Sistem. Proses ini merupakan pendekatan bersistematik untuk analisa dan rekebentuk sistem maklumat. Terdapat beberapa langkah utama yang perlu dijalankan iaitu :-

- Menenalpasti masalah, dan peluang
- Menentukan keperluan maklumat
- Menganalisis keperluan maklumat
- Merekebentuk sistem yang dicadangkan
- Membangunkan dan mendokumentasi perisian
- Menguji dan menyelenggara sistem
- Mengimplementasi dan menilai sistem

Perancangan merupakan strategi yang terpenting didalam membangunkan sistem ini bagi menepati objektif yang telah digariskan. Langkah pertama yang akan diambil ialah mencari maklumat seberapa banyak yang mungkin. Banyak masa diberikan kepada pengumpulan bahan, menganalisa maklumat dan juga merekebentuk sistem yang akan dibangunkan kelak.

Mempelajari perisian yang akan digunakan juga penting kerana kecekapan di dalam membuat pengaturcaraan akan memberikan kesan kepada hasil sistem tersebut.

Peringkat pengujian dijalankan di akhir pembangunan dan sebarang perubahan akan dibuat semasa peringkat ini.

Projek ini dimulakan pada semesta I sesi 2002/2003. Tarikh pengesahan tajuk ini pada 8 jun 2002. Projek ini dibahagikan kepada dua fasa dimana fasa pertama

dilaksanakan pada semesta I sesi 2002/2003 dan fasa kedua dilaksanakan pada semesta II sesi 2002/2003.

Fasa pertama melibatkan kajian awal, analisa sistem, dan rekebentuk sistem. Kajian awal bermula pada 15 Julai 2002 dan berakhir 26 Ogos 2002. Analisa projek dimulakan pada 12 Ogos 2002 dan berakhir pada 9 September 2002. rekebentuk sistem imej kepada data ini pula akan direka pada 26 Ogos 2002 dan berakhir pada 23 September 2002.

Fasa kedua melibatkan proses pengkodan, pengujian, pelaksanaan, dan penyelenggaraan projek. Proses pengkodan dan pengujian akan dimulakan bagi membuka tirai fasa kedua. Kemudian proses pelaksanaan dan penyelenggaraan akan menyusul. Hanya anggaran tarikh yang dijangkakan dibuat untuk fasa kedua ini kerana tarikh yang tetap untuk menjalankan proses-proses tersebut masih tidak dapat ditentukan lagi.

Walaupun bagaimanapun, jadual perancangan projek dibawah dapat menggambarkan secara anggaran penjadualan bagi keseluruhan projek.

Jadual 1.1: Jadual Perancangan Projek

Jadual	15/7/02	29/7/02	12/8/02	26/8/02	9/9/02	23/9/02	7/10/02	21/10/02	4/11/02	18/11/02	2/12/02	16/12/02	30/12/02	6/1/03	20/1/03	3/2/03	17/2/03	3/3/03
Kenalpasti masalah dan peluang	■	■																■
Menentukan keperluan maklumat		■	■															■
Menganalisa keperluan maklumat			■	■														
Mencadangkan rekebutuk sistem				■	■													
Pembangunan sistem						■	■	■	■	■	■							
Ujian unit dan gabungan											■	■	■					
Menguji sistem														■	■			
Ujian kebolehterimaan																■		
Operasi dan penyelenggaraan																	■	■

2.0 KAJIAN LITERASI

2.1 Definisi Sistem

Maksud imej kepada data adalah menukarkan grafik imej kepada perwakilan data. Data mengikut makna dalam kamus dewan (edisi baru), 1989 bermaksud butir-butir atau maklumat terperinci yang diketahui atau telah dikumpulkan tentang sesuatu dan dapat dijadikan asas untuk membuat kajian (analisis atau kesimpulan). Terdapat banyak jenis data digunakan dalam masa sekarang dan jenis data yang dipilih untuk sistem adalah data binari. Data binari sangat luas penggunaannya dan ia adalah dalam skop bidang komputer yang terlibat. Imej disini mewakili ilustrasi grafik yang berkonsepkan memaparkan bentuk gambaran bukan bermaksud keperibadian atau watak seseorang atau sesuatu. Contoh imej yang dimaksudkan adalah seperti di bawah :-



Gambarajah 2.1 : Contoh imej

2.2 Kajian Terhadap Sistem Yang Telah Wujud

Sistem Pemprosesan Borang Genie

Sistem Pemprosesan Borang Genie mengimbas borang kertas dan ditukarkan terus kepada data sekaligus mengurangkan masa kemasukan data dan kos. Genie adalah satu modul. Aliran kerja sistem Genie berintegrasi dengan pengimbas, giliran, server pemprosesan borang, stesen ubah atau pembetulan, dan server eksport imej dan data kepada persekitaran pemprosesan selari yang sebenar. Genie adalah fleksibel berfungsi dengan semua komponen pada stesen tunggal atau diagih kepada workstation dan server. Pejabat workstation boleh berautomatik sambung kepada aliran kerja Genie dengan mengaktifkan komponen pada stesen itu dimana ia membenarkan skedul fleksibel.

Kemodulan juga bermaksud banyak bahagian borang dalam persekitaran aliran kerja yang berskala terus kepada bilangan kerja yang telah dilakukan. Jika semakin tinggi bilangan bertambah, hanya perlu buat tambahan server dan workstation untuk uruskan beban tambahan itu. Pengguna hanya perlu membayar sebanyak mana kuasa yang pengguna perlukan dan membuat tambahan hanya apabila pengguna perlukan sahaja.

Sistem juga mempunyai stesen pengimbas untuk menangkap imej, server baca borang untuk ditukarkan imej kepada data, stesen ubahsuai untuk membaiki kesalahan atau sebagai kunci dari imej, dan stesen eksport untuk pindah masuk data kepada pangkalan data dan indeks imej untuk pangkalan data imej.

Imej yang diimbas oleh stesen pengimbas diletakkan ke dalam bekas objek dan dihantar kepada aturan imbas. Setiap titik proses dalam aliran kerja berfungsi seperti satu kilang, menerima bekas bahan mentah daripada aturan input, buka bekas itu, mengkaji kandungan didalamnya, ubahsuai sesetengah kandungan tersebut, masukkan bahan baru kedalam bekas dan akhir sekali hantar barang tersebut kepada aturan output untuk tunggu proses yang seterusnya. Apabila semua proses telah siap, hasil produk akan dihantar kepada pengguna sebagai data dan imej.

Proses-proses sistem Genie

Semua proses Genie adalah program bebas jalan (independent executable). Apabila dijalankan ia mengambil ruang prosesnya sendiri, tidak memberi kesan secara terus dengan proses jalan sendiri yang lain. Setiap proses berkomunikasi dengan proses lain melalui Pesanan Rangkaian Kerja STT. Terdapat lima jenis proses iaitu :-

1. Proses pesanan.

Standard proses pesanan TXP menukarkan pesanan antara proses pada komputer tunggal. Proses pesanan LXP memantau pesanan antara proses pada komputer berbeza dalam rangkaian komputer LAN.

2. Proses aturan.

Satu aturan menyimpan objek dalam susunan tarikh dan masa sehingga mereka dipanggil oleh proses lain.

3. Proses input.

Proses input mewujudkan satu bekas objek baru dan masuk data kedalamnya dan dihantar ke aturan input. Stesen imbas adalah pemproses input.

4. Proses kerja.

Proses kerja mengambil bekas objek dari satu aturan kemudian diperiksa, diubahsuai, ditambah atau dihapuskan data dari objek dan dihantar objek kepada aturan yang lain. Server pemprosesan borang dan server ubahsuai adalah pemproses kerja.

5. Proses output

Proses output dapatkan bekas objek dari aturan eksport kemudian dibuka, keluarkan data yang penting bagi pengguna, diletakkan data ke dalam borang pengguna dan akhir sekali objek dihapuskan. Stesen eksport adalah pemproses output.

Komponen Genie

Bekas Objek

Bekas objek adalah seperti analog penghantaran karton yang penuh dengan bungkusan kecil. Ia digunakan untuk mengumpul semua bungkusan data, imej dan kawalan ke dalam unit penghantaran tunggal. Setiap bungkusan objek mengandungi satu unit maklumat tentang kepentingan kepada sistem seperti pemampatan halaman imej, hasil daripada pati bacaan borang, atau maklumat kawalan.

Penandaan Tarikh dan Masa, serta Sumber Imej

Setiap bekas objek diberikan sistem pengesahan unik apabila dibuat oleh proses input. Id bergantung kepada penanda tarikh dan masa, dan id tempat dimana objek direka. Id ini kekal malar sepanjang objek tersebut masih ada.

Aturan

Aturan adalah program bebas (independent) yang berkomunikasi dengan proses aktif melalui pesanan. Aturan direka hanya untuk memegang bekas objek dalam susunan penanda tarikh dan masa sehingga mereka dipanggil. Aturan boleh menerima objek dari sebarang aliran proses pada bila-bila masa. Aturan akan memberi tindak balas aliran proses meminta objek dengan sama ada objek ada lagi atau tidak.

Stesen Pengimbasan

Stesen pengimbasan adalah program proses input yang menerima dan membungkus satu atau lebih imej dari pengimbas atau sumber input lain ke dalam bekas objek dan hantarnya ke aturan imbas. Proses input bertanggungjawab untuk menambah penanda tarikh dan masa serta sumber id kepada bekas objek.

Server Pemproses Borang

Server pemproses borang adalah program proses kerja yang mengambil bekas objek dari satu atau lebih aturan input, keluarkan imej, proseskan setiap imej yang dijumpai melalui server borang, letak hasil dari setiap imej kembali ke bekas objek dan akhir sekali hantar bekas objek ke aturan destinasi yang bergantung kepada keadaan set bekas objek. Kelaziman destinasi aturan output untuk server pemproses borang adalah aturan eksport.

Stesen Ubah Data atau Baiki

Stesen ubah data atau baiki adalah program proses kerja yang mengambil bekas objek dari satu atau lebih aturan ubah atau baiki, keluarkan imej, posisikan operator pada ralat data dengan memaparkan posisi yang betul. Operator membaiki hanya pada kawasan ralat. Apabila telah siap, operator hantar bekas objek yang lengkap ke aturan destinasi dan menjemput bekas objek yang seterusnya. Kelaziman destinasi aturan output untuk server pemproses borang adalah aturan eksport.

Stesen Eksport

Stesen eksport adalah program proses output yang mengambil bekas objek dari satu atau lebih aturan eksport, keluarkan imej, gunakan skrip pengguna untuk format timbunan data dalam data dan fail imej. Apabila bergabung dengan sistem hubungan imej yang betul, stesen eksport boleh automatik pindah masuk indek imej dan sebahagian imej (tandatangan atau sub-imej) kepada pangkalan data imej.

2.3 Persekitaran Teknologi Pengaturcaraan

Microsoft Visual C++

Microsoft Visual C++ adalah perisian pembangunan yang menggunakan bahasa pengaturcara C++. C++ telah menggantikan zaman bahasa C yang telah lama wujud tetapi bahasa C masih termasuk dalam bahasa C++. Tambahan pula C++ telah mengaplikasi teknik terkini pengaturcaraan iaitu pengaturcaraan berorientasikan objek. Kelebihan bahasa C++ adalah ia masih mengandungi bahasa C dan bahasa berorientasikan objek.

Perisian ini juga menyediakan kemudahan perpustakaan (DLL) yang ada fungsi terbina dalam (built-in function). Ia juga membenarkan pengaturcaraan memilih jenis hubungan semasa pindah muat atau hubungan semasa dijalankan dalam perpustakaan.

Persembahan berkonsepkan pengaturcaraan berorientasikan objek (OOP) yang mempunyai kelebihan dalam teknik kelas, perwakilan, dan polimorfisme yang membolehkan tugas-tugas pembangunan diguna semula dan dilanjutkan dengan lebih cekap.

Disebabkan C++ adalah standard, perkakasan sendirian (hardware-independent), bahasa penggunaan meluas, aplikasi yang ditulis dalam C++ selalunya boleh dilarikan dengan sedikit atau kurang pengubahsuaian pada pelbagai kategori dalam komputer yang berlainan. Maka C++ diterima pakai oleh umum.

2.4 Persekitaran Sistem Pengendalian

Komputer terdahulu hanya berkemampuan melakukan satu tugas atau kerja pada satu-satu masa. Bentuk operasi komputer sebegini selalunya dipanggil pemprosesan kumpulan sekali guna. Komputer berfungsi satu program pada satu masa ketika memproses data dalam kumpulan. Dalam sistem terdahulu ini, pengguna biasanya melakukan kerja pada komputer pusat menggunakan kad tebuk. Pengguna selalu terpaksa menunggu berjam-jam atau sehari sebelum hasil cetak dikeluarkan.

Perisian sistem yang dipanggil sistem pengendalian dibangunkan untuk membantu memudahkan menggunakan komputer. Sistem pengendalian terdahulu mampu melicinkan transaksi antara kerja. Ini mengurangkan masa diambil oleh operator komputer untuk penukaran antara kerja dan menambah jumlah kerja komputer dalam proseskan.

Dari masa ke semasa komputer menjadi lebih berkuasa tinggi dan sistem pengendalian mengalami revolusi peningkatan kuasa pemprosesan. Salah satu sistem pengendalian yang popular digunakan iaitu Microsoft Windows maju dari setingkat ke tingkat ke hadapan seiring dengan peredaran zaman. Bermula dengan versi perisian Windows 1.0 meningkat ke Windows 95, Windows 98, Windows 2000, Windows ME, dan terbaru sekali Windows XP. Kemampuan perisian Windows ini amat mengagumkan baik dalam apa jua lapangan bidang penggunaan sistem komputer. Selain sistem pengendalian Microsoft Windows terdapat sistem pengendalian dalam pasaran kini iaitu Linux. Linux juga mendapat sambutan penggunaan terbesar kedua selepas Microsoft Windows. Sistem Linux tidak jauh berbeza berbanding Windows kerana ciri-cirinya adalah lebih kurang sama sahaja. Satu daripada perbezaan kedua-dua sistem ini ialah Linux boleh didapati secara percuma dan perisian Windows perlu dibeli. Sistem Linux berasal daripada sistem Unix yang juga sistem pengendalian. Unix banyak digunakan pada suatu ketika dahulu.

2.5 Kajian Teknik Pengecaman Objek Yang Digunakan

Pengesanan penjuru menggunakan pendekatan spatial

Menggunakan ciri spatial pada penjuru boleh bersumber suatu nombor pendekatan untuk pengesanan penjuru. Untuk memahami perkara asas tentang teknik ini, boleh dimulakan dengan memerhati jadual jujukan nombor berikut :-

Jadual 2.1 : Jujukan nombor

-1	2	-1
-1	2	-1
-1	2	-1

Dianggapkan imej mempunyai hanya paras dua gray, 0 dan 1, dan ketegakan penjuru paling kurang tiga pixel panjang wujud di sesuatu tempat dalam imej. Untuk mengesan seperti penjuru tegak, kita imbas jujukan nombor dalam imej. Pada setiap lokasi kita darabkan kandungan jujukan nombor dengan paras gray yang sepadan dan jumlahkan hasilnya. Oleh kerana jumlah semua elemen jujukan nombor bersamaan sifar, hasil adalah sifar bagi latarbelakang seragam dan bukan sifar bagi selainnya. Jika elemen baris tegak ditengah jujukan nombor sama nilai dengan elemen baris tegak suatu penjuru paling kurang tiga pixel dan dikanan bernilai 1 serta dikiri bernilai 0, kita akan peroleh hasil ialah 3 seperti berikut :-

$$(-1 \times 0) + (-1 \times 0) + (-1 \times 0) + 2 \times 1 + 2 \times 1 + 2 \times 1 + (-1 \times 1) + (-1 \times 1) + (-1 \times 1) = 3$$

Pengiraan di atas boleh diringkaskan dengan persamaan berikut :

$$y(n_1, n_2) = \sum_{k_1=-1}^1 \sum_{k_2=-1}^1 h(k_1, k_2) I(n_1 + k_1, n_2 + k_2) \quad (2.0)$$

Dimana $h(k_1, k_2)$ adalah elemen jujukan nombor, dengan $h(0,0)$ adalah tengah-tengah elemen, dan $I(n_1, n_2)$ adalah paras (intensity) imej. Jelas bahawa dari perbincangan di atas yang maklum balas maksimum langkah diatas akan paling kurang tiga piksel tinggi pada penjuru tegak. Persamaan (2.0) merujuk kepada *cross correlation* jujukan nombor dengan imej. Jujukan nombor selalunya dirujuk sebagai operator penjuru. Terdapat dua operator yang selalu digunakan iaitu operator Roberts dan operator Sobel. Saya gunakan operator Sobel kerana lebih sesuai berbanding operator.

Operator Sobel

Operator Sobel direka untuk hampir sama dengan fungsi kecerunan diskrit. Operator Sobel adalah seperti berikut :-

W_1

1	0	-1
2	0	-2
1	0	-1

W_2

-1	-2	-1
0	0	0
1	2	1

Kekuatan penjurur imej boleh dihitung dengan menggunakan persamaan seperti berikut :

$$y(i,j) = \sqrt{y_1(i,j)^2 + y_2(i,j)^2}$$

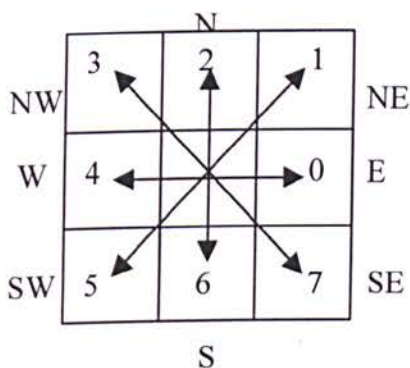
Arah penjurur boleh dihitung dari persamaan di bawah :

$$\Theta(i,j) = \tan^{-1} \left[\frac{y_2(i,j)}{y_1(i,j)} \right]$$

W_1 adalah untuk mengesan penjurur tegak dan W_2 adalah untuk penjurur datar. Kegunaan lebar sifar sepanjang penjurur adalah untuk menindas kegugupan.

Operator Kompas Kecerunan

Operator kompas kecerunan direka melalui pertimbangan semua orientasi yang berkemungkinan penjurur dalam imej diskrit. Oleh itu, daripada menggunakan dua operator sahaja, lapan operator digunakan boleh memudahkan proses. Setiapnya membekalkan kekuatan penjurur bersama satu daripada lapan kemungkinan arah kompas(lihat gambarajah 2.2 : Arah Kompas). Ini direka berdasarkan asas model data untuk penjurur dalam imej.



Gambarajah 2.2 : Arah Kompas

3.0 METODOLOGI

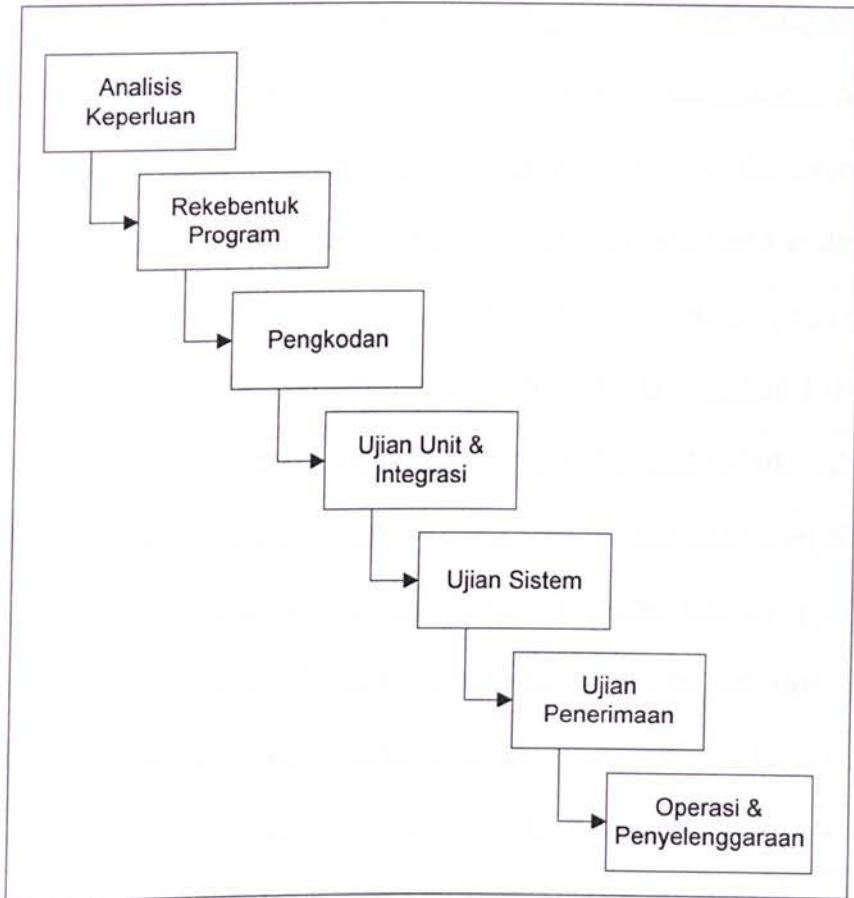
3.1 Pengenalan

Pelbagai metodologi yang dapat digunakan untuk suatu proses pembangunan perisian. Metodologi adalah susunan model pembangunan perisian yang digunakan bersama-sama dengan satu atau lebih teknik pembangunan perisian. Metodologi yang bersesuaian dan tepat memainkan peranan penting bagi penghasilan dan persembahan produk perisian atau sistem yang boleh dipercayai dan betul. Metodologi yang telah dipilih mestilah menepati ciri-ciri domain masalah yang sebenar.

Terdapat dua objektif metodologi pembangunan sistem iaitu :-

1. Memastikan semua individu yang terlibat dalam projek sedia maklum dengan tujuan, progres dan masalah bagi suatu pembangunan perisian.
2. Membahagikan pembangunan projek kepada beberapa langkah yang fasa-fasa yang boleh diuruskan dengan penyempurnaan berisyarat oleh hasil akhir yang spesifik.

3.2 Model Air Terjun (Waterfall Model)



Gambarajah 3.1 : Model Air Terjun

Pendekatan Pembangunan

Model airterjun didapatkan daripada model kejuruteraan untuk mengatur proses pembangunan suatu produk perisian yang besar. Ia mengandungi beberapa tahap yang berbeza yang diproses secara linear. Berbanding model pembangunan yang lain, ia lebih teguh dan boleh diuruskan dengan lebih baik. Ia sejenis model yang penting dimana menjadi asas kepada pembentukan model-model lain. Walau bagaimanapun, kebanyakan projek yang lebih moden, model airterjun ini agak ketinggalan zaman. Model ini masih lagi digunakan secara meluas.

Pada permulaannya, pengaturcaraan perisian adalah mudah dan dilakukan oleh seorang pengaturcara dan digunakan untuk tujuan saintifik dan kejuruteraan.

Walaupun begitu, seiring dengan perkembangan penggunaan komputer yang berkembang pesat, perisian perlu ditulis untuk pengguna lain bagi mereka yang kurang pengetahuan tentang pengaturcaraan. Contohnya, perisian yang berkaitan dengan perakauan tetapi tidak mempunyai pengetahuan tentang pengaturcaraan. Oleh itu, perisian yang mereka gunakan tidak seharusnya membebankan dengan keperluan bahasa pengaturcaraan. Idea lama dalam menulis suatu program dan kemudian mengatasi masalah pipit tidak lagi memadai. Dalam tahun 1970, Royce telah menyusulkan suatu model untuk pembangunan perisian, diterbitkan daripada model yang sama digunakan dalam aktiviti kejuruteraan. Anggapan yang dibuat pada masa itu adalah pembangunan perisian pembangunan perisian ialah disiplin kejuruteraan. Oleh itu, ia akan menuruti suatu model. Model ini telah diterima dan digunakan dengan dikenali sebagai model airterjun. Tidak beberapa lama kemudian, didapati bahawa model airterjun ini hanya sesuai dipraktikkan dan berkesan kepada beberapa klas perisian yang lebih kompleks telah banyak diwujudkan. Model asal oleh Royce telah diperbaiki dan diubahsuai sedikit demi sedikit mengikut perkembangan semasa.

Idea yang digunakan dalam pembangunan sistem menggunakan model airterjun ini ialah perbezaan beberapa peringkat yang terdapat dalam proses pembangunannya. Setiap peringkat mempunyai input yang akan menghasilkan output. Output ini akan digunakan oleh peringkat yang seterusnya. Pada kebiasaannya, terdapat tujuh peringkat dalam model pembangunan sistem ini.

1. Analisa keperluan dan penerangan

Pada peringkat ini, keperluan perisian yang akan dibangunkan dibentuk.

Keperluan yang dimaksudkan ialah perkhidmatan yang ia akan berikan, kekangan dan sasaran perisian tersebut. Sebaik sahaja ia diwujudkan ia harus ditaksirkan supaya ia boleh digunakan dalam peringkat seterusnya. Peringkat ini kerap dimulakan dengan suatu kajian ketersauran atau suatu penelitian kemungkinan dimasukkan dalam peringkat ini. Kajian kesauran mengandungi persoalan seperti:-

- Perlukah perisian dibangunkan?
- Adakah alternatif lain?

Ia boleh dikatakan konsep produk perisian dan berkemungkinan dilihat sebagai permulaan bagi kitar hayat pembangunan.

2. Rekebentuk sistem dan perisian

Pada peringkat ini, keperluan yang telah wujudkan daripada peringkat pertama dikenalpasti sebagai keperluan perkakasan atau perisian. Keperluan perisian kemudiannya diterjemahkan kepada bentuk yang sedia ditukarkan kepada program komputer.

3. Pengkodan

Pengaturcara akan menjalankan pembangunan perisian membuat aturcara kod bagi mencipta perisian. Program-program akan dibina berdasarkan keperluan berbuat demikian. Jangkaan dan anggaran kejayaan program perisian yang dihasilkan akan ditentukan selepas program siap dibina.

4. Ujian Unit dan Integrasi

Peringkat ini pula merupakan suatu peringkat dimana program perisian akan diuji. Setiap program dikenali sebagai unit dan ujian unit ialah pengesahan sama ada unit tersebut menepati spesifikasi yang telah disenaraikan sebelum ini.

5. Ujian Sistem

Semua unit digabungkan dan keseluruhan sistem diuji. Apabila gabungan program berjaya dalam pengujian ini, produk perisian ini telah lengkap dan selesai.

6. Ujian Penerimaan

Sistem diuji dengan keadaan persekitaran yang sebenar. Sistem akan dipasang pada berbilang versi sistem pengendalian untuk menentukan masalah yang wujud sekiranya ada. Imej pelbagai jenis bentuk dan saiz objek diujikan ke atas sistem. Format imej juga dipilih berbeza-beza untuk kebolehan sistem berfungsi untuk pelbagai sumber format.

Sistem akan dipersembahkan kepada pengguna untuk kelulusan penggunaan. Pengguna akan meminta perubahan atau pembetulan jika ada masalah yang wujud atau ciri-ciri kehendak pengguna tidak lengkap didalam sistem. Pereka bentuk sistem akan merancangkan penambahbaikan berdasarkan hasil perbincangan dengan pengguna.

7. Operasi dan Penyelenggaraan

Kebanyakan pembangunan perisian mengandungi peringkat ini. ia melibatkan pembetulan kesalahan yang tidak dapat dikesan sebelum ini. selain itu, dalam

peringkat ini juga perisian diperbaiki dan pelbagai sokongan lain diberikan. Ia merupakan sebahagian daripada kitar hayat produk perisian. Menuju perubahan ke arah yang lebih baik dan pembaikan dianggap sebagai 'pembangunan'.

Langkah-langkah di atas merupakan peringkat utama dalam model airterjun.

Terdapat sub-peringkat lain dalam setiap peringkat tetapi berbeza dari suatu projek ke suatu projek yang lain. Contohnya, bagi tujuan pengurusan peringkat keperluan dibahagikan kepada kajian kesauran penentuan keperluan garis kasar, kajian rekebentuk dan peringkat spesifikasi keperluan.

Ia berkemungkinan juga bagi sesetengah projek pembangunan perisian memerlukan peringkat tambahan ataupun memecahkan satu peringkat kepada dua peringkat yang berbeza. Walaupun begitu, semua model airterjun yang berbeza mempunyai idea yang terpendam yang sama iaitu suatu peringkat membekalkan peringkat yang seterusnya dengan output yang digunakan sebagai input. Dengan demikian terdapat suatu aliran linear di antara peringkat-peringkat tersebut.

Akhir sekali, perlu diambil kira bahawa proses pembangunan perisian tidak selinear yang dilihat. Apabila kesalahan dikesan dalam peringkat yang bawah, ia akan kembali kepada peringkat yang sebelumnya untuk membetulkan kesilapan tersebut. Memandangkan ia merupakan beban pengurusan, seringkali kesalahan dan masalah tidak dihiraukan atau dibiarkan untuk tindakan selanjutnya kemudian, kesan kepada model airterjun ini dengan maklumat dapat mengalir dua hala, ke bawah apabila sesuatu telah dibuat dan ke atas apabila terdapat kesalahan atau maklumbalas diberikan. Di samping itu, proses akan dibekukan jika belum tiba masa untuk

diuruskan. Ini membawa kepada pembentukan model-model lain yang lebih fleksibel.

4.0 ANALISA SISTEM

Analisa sistem yang dikaji adalah mengenai struktur dalaman sistem iaitu keperluan sistem. Ini merangkumi keperluan fungsian, keperluan bukan fungsian, keperluan perisian, dan keperluan perkakasan.

4.1 Keperluan Fungsian

Keperluan fungsian menerangkan interaksi diantara sistem dan juga persekitarannya.

Ia menjangkakan bagaimana sesuatu sistem akan bertindak pada sesuatu keadaan.

Berikut di bawah adalah senarai keperluan fungsian :-

- Mengesan objek daripada imej
- Mendapatkan data dalam bentuk binari daripada objek
- Paparkan data binari objek kepada pengguna
- Mengekstrak pengetahuan daripada data
- Format imej yang berlainan boleh diproses

4.2 Keperluan Bukan Fungsian

Keperluan bukan fungsian menerangkan kekangan ke atas sistem yang menyebabkan pilihan dalam membangunkan penyelesaian terhadap masalah dihadkan. Berikut

adalah senarai keperluan bukan fungsian :-

- Antaramuka yang mesra pengguna
- Informasi yang jelas dan mudah pengguna
- Masa tindakbalas yang pantas

4.3 Pertimbangan- pertimbangan Lain

4.3.1 Keperluan Perisian

- Microsoft Visual C++
- Microsoft 98/ME/2000/XP sistem pengendalian

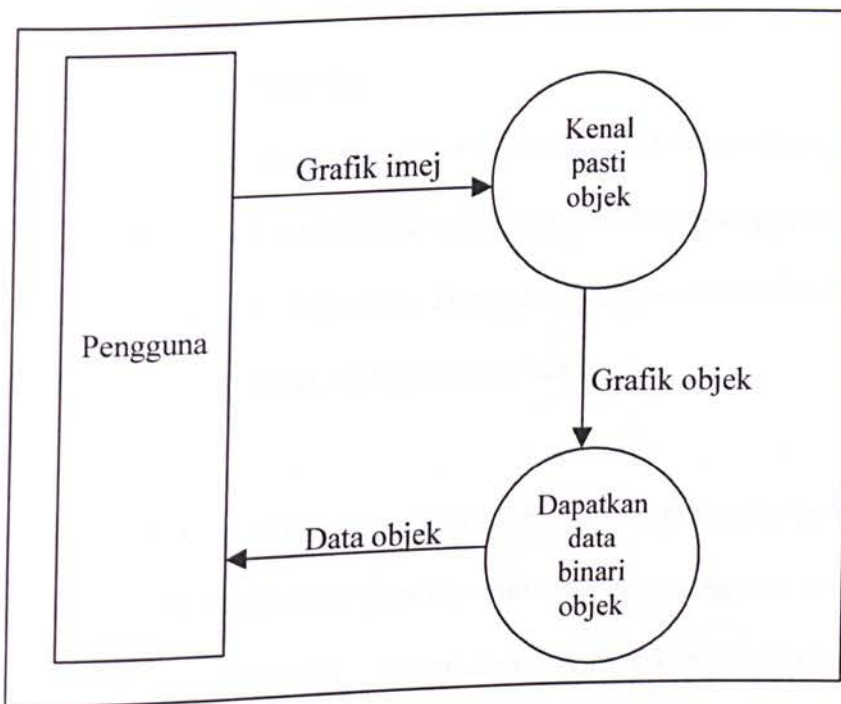
4.3.2 Keperluan Perkakasan

- Sebuah komputer peribadi
- Ruang storan yang kecil (anggaran saiz data 50 MB)
- Sekurang-kurangnya 64MB RAM memori
- Unit pemprosesan pusat (CPU) berkuasa 300Mhz paling minimum

5.0 REKEBENTUK SISTEM

5.1 Carta Aliran Data

Pengguna memberikan sumber kepada sistem dalam bentuk fail imej dan kemudian sistem akan memaparkan imej yang terdapat didalam fail tersebut. Sistem akan mula menjalankan proses mengecam objek yang wujud dalam imej, objek dapat dikenalpasti, grafik objek tersebut akan dijalankan proses mendapatkan data binari daripada objek itu. Hasil data binari yang diperolehi daripada objek itu akan dipaparkan kepada pengguna melalui skrin sistem.



Gambarajah 5.1 : Carta Aliran Data

5.2 Rekebentuk Senibina

Rekebentuk senibina menyatakan struktur keseluruhan bagi sistem perisian. Ia adalah peringkat awal bagi proses rekebentuk sistem. Tujuan utama adalah mengenalpasti

komponen sistem utama dan komunikasi sesamanya. Di bawah adalah rekebentuk sistem imej kepada data :



Gambarajah 5.2 : Senibina sistem imej kepada data

5.3 Rekebentuk Antaramuka

Rekebentuk antaramuka adalah pusat utama bagi hubungan di antara pengguna dan sistem komputer. Ianya adalah bahagian sistem di mana pengguna nampak, dengar, sentuh dan berhubung dengannya. Pengguna sentiasa berinteraksi dengan sistem komputer bagi melaksanakan tugas-tugas mereka.

Kebanyakan sistem hari ini menggunakan antaramuka pengguna bergrafik (GUI). Ianya mudah dipelajari dan digunakan walaupun bagi pengguna yang kurang biasa dengan sistem komputer. Oleh itu untuk sistem imej kepada data ini, saya menggunakan konsep antaramuka pengguna bergrafik. Prinsip antaramuka sistem ini adalah seperti diterangkan di bawah ;-

- Kebiasaan pengguna

Istilah dan konsep yang biasa kepada pengguna dicerapkan ke dalam sistem dan bukannya mengikut apa yang biasa kepada perekebentuk.

- Kekonsistenan

Kekonsistenan dapat mengurangkan masa mempelajari sistem. Arahkan, menu dan ikon perlu mempunyai format yang sama bagi keseluruhan sistem.

- Perbezaan minimum

Arahan mesti sama dan mempunyai kesan atau perlakuan yang sama.

- Kebolehpulihan

Antaramuka akan mengandungi kemudahan yang dapat membantu pengguna pulih dari kesilapan yang dibuat. Contohnya adalah kemudahan 'undo'.

- Panduan Pengguna

Kemudahan bantuan dan sebagainya akan disediakan untuk kegunaan pengguna. Sokongan pengguna seperti mesej ralat dan dokumentasi pengguna juga memudahkan pengguna. Contohnya adalah kemudahan 'Help' yang boleh memberi maklumat lanjut mengenai ralat dan maklumat pengetahuan tentang sistem perisian.

Ciri-ciri antaramuka sistem adalah menggunakan tettingkap, menu, ikon, penunjuk, dan elemen grafik. Berikut jadual di bawah menerangkan tentang ciri-cirinya :-

Ciri	Penerangan
Tetingkap	Berbilang tingkap membenarkan maklumat yang berbeza dipaparkan serentak pada skrin pengguna
Ikon	Ikon boleh membawa mewakili maklumat. Pada sesetengah sistem, ikon mewakili fail, dan sesetengahnya mewakili proses.
Menu	Arahan dipilih dari menu berbanding menulis arahan dalam bahasa arahan.
Penunjuk	Alat penunjuk seperti tetikus digunakan untuk memilih pilihan dari menu atau bahagian tertentu dalam tettingkap
Grafik	Elemen grafik boleh bergabung dengan tulisan pada paparan yang sama.

Jadual 5.1 : Ciri-ciri antaramuka

6. PELAKSANAAN SISTEM

Pelaksanaan sistem atau implementasi sistem dijalankan dengan merujuk kepada rekabentuk sistem yang disediakan pada fasa analisis dan rekabentuk. Rujukan ini amat penting untuk memastikan pembangun sistem mematuhi segala keperluan yang harus wujud dalam sistem tersebut. Oleh yang demikian, perkara utama dan terpenting dalam fasa ini adalah pengodan yang merupakan senarai susunan set aturcara yang melarikan program.

6.1 Pengkodan Sistem

Pengkodan sistem ini banyak melibatkan daya kreativiti bagi pengaturcara.

Rekabentuk adalah panduan untuk fungsi atau tujuan sesuatu komponen dalam sesuatu sistem. Perancangan rekabentuk yang tersusun mampu menghasilkan suatu pengkodan yang menarik. Ini dapat diperhatikan kepada pengaturcara yang ada fleksibiliti dalam mengimplimentasikan rekabentuk kepada kod. Rekabentuk dan speksifikasi keperluan mungkin akan mencadangkan bahasa pengaturcaraan.

Dalam aspek pengkodan sistem, setiap komponen program melibatkan tiga aspek utama iaitu,

- Struktur kawalan
- Algoritma
- Struktur data

6.1.1 Struktur Kawalan

Kebanyakan struktur kawalan untuk komponen dicadangkan dalam senibina dan rekabentuk, maka ia akan diterjemahkan kepada kod. Tidak kira apa jenis rekabentuk, struktur program mesti mencerminkan rekabentuk struktur

kawalan. Banyak panduan dan piawai yang mencadangkan kod ditulis dengan cara dimana pengaturcara mudah membaca komponen dari atas ke bawah. Struktur kawalan haruslah mudah supaya pengguna tidak tersesat semasa membaca kod.

6.1.2 Algoritma

Kita mesti menyeimbangkan masa yang diterima dengan kualiti rekabentuk, piawaian dan keperluan pengguna. Untuk sistem ini, banyak kod ditulis dan fungsi digunakan. Pengkodan banyak dihasilkan bergantung kepada objektif fungsi-fungsi itu. Lebih besar objektif sesuatu fungsi itu, maka banyak kod dituliskan. Program ini boleh dikatakan mempunyai kurang bilangan fungsi berikutan ciri-ciri keperluan fungsian sistem tidak banyak bilangannya. Namun ada sebilangan keperluan fungsian itu membentuk fungsi yang besar.

6.1.3 Struktur Data

Dalam menulis program, kita mesti format dan simpan data dan juga manipulasi data adalah mudah.

6.1.4 Alat Pembangunan

Keperluan perkakasan

- Pemproses pentium 1/ II/ Celeron/ III/ IV
- 64 MB Memori
- Kapasiti cakera keras 200MB
- Monitor

- Papan kekunci dan tetikus

Alatan pembangunan ini dinilai dari segi setakat mana prestasinya dapat membantu keperluan fungsian. Adakah ia mudah digunakan dan kemudahan untuk menyediakan dokumentasi?. Pemilihan yang tepat dapat menjamin kesempurnaan perlaksanaan dalam fasa ini berkait tepat dengan alatan pembangunan sistem atau perisian.

Beberapa kriteria penting yang dipertimbangkan dan diberi penekanan dalam pemilihan alatan pembangunan yang bersesuaian ialah:

- Bahasa pengaturcaraan yang dipilih perlu menyokong teknik manipulasi imej, fail, dan teks.
- Bahasa pengaturcaraan yang mampu memberikan kemudahan untuk rekabentuk antara muka pengguna yang bercirikan interaktif dan menarik.
- Bahasa yang digunakan adlah terkini dan sejajar dengan perkembangan teknologi.
- Kemudahan sistem pengoperasian pada persekitaran di mana bahasa pengaturcaraan tersebut boleh dilarikan.
- Ciri-ciri istimewa dan fungsi utama yang mampu ditawarkan oleh alatan pembangunan tersebut.
- Mudah digunakan sama ada dari aspek sintetik dan logikal.
- Wujudnya kemudahan pengesanan dan pengendalian kawalan ralat dan pepijat.
- Pengetahuan asas terhadap konsep dalam bahasa pengaturcaraan tersebut.

Untuk memenuhi kriteria-kriteria yang digariskan, maka Microsoft Visual C++ versi 6.0 dipilih sebagai alatan pembangunan sistem ini. Microsoft Visual C++ adalah persekitaran yang baik untuk pembangunan perisian windows dengan teknologi terkini. Ianya merupakan antara perisian pembangunan multimedia yang mudah untuk digunakan dan juga merupakan salah satu perisian yang hebat. Pengaturcara tidak perlu menjadi pakar untuk membina sistem yang canggih dan menarik kerana pembelajaran bagi sistem boleh dibuat semasa proses pembangunan. Microsoft Visual C++ versi 6.0 menyediakan kemudahan “template” binaan dalaman untuk membina kerangka asas aplikasi dari penggunaan *Microsoft Foundation Class* (MFC) *Library* hingga *ActiveX*. Dalam MFC banyak kelas-kelas disediakan bagi memudahkan pengaturcara membuat pelbagai pengaturcaraan berbagai jenis sokongan. Ini membuktikan Microsoft Visual C++ memang perisian yang sesuai dan tepat.

6.1.5 Alatan Grafik

6.1.5.1 Adobe Photoshop 7.0

Saya telah menggunakan perisian ini untuk mereka imej bagi tujuan pengujian sistem. Kriteria imej yang dihasilkan adalah pelbagai objek imej yang berbeza-beza saiz, bentuk dan warna. Ini adalah penting untuk memenuhi proses pengujian sistem.

6.2 Dokumentasi Program

Dokumentasi program merupakan satu set penerangan bertulis yang menerangkan kepada pengguna apa yang program itu akan lakukan dan bagaimana ia akan dilaksanakan. Ia terbahagi kepada dua iaitu dokumentasi dalaman dan dokumentasi luaran. Dokumentasi dalaman merupakan material penerangan yang bertulis secara terus kepada kod. Dokumentasi yang lain ialah dokumentasi luaran.

6.2.1 Dokumentasi Dalaman

Dokumentasi dalaman dimasukkan sekali ke dalam kod aturcara. Ini bagi menerangkan maksud atau objektif bagi bahagian tertentu aturcara. Kaedah ini memudahkan pengaturcara atau individu bebas mengetahui perkara mengenai bahagian itu. Pengubahsuaian atau guna semula aturcara dapat dilakukan dengan mudah dan berkesan.

6.2.2 Dokumentasi Luaran

Dokumentasi program adalah satu set penerangan bertulis yang menerangkan kepada pembaca apa yang program tersebut akan lakukan dan bagaimana ia dilakukan.

Setiap kerangka mempunyai fungsi, subfungsi, pengistiharaan untuk pembolehubah yang digunakan untuk fungsi dan juga untuk membina sebuah tayangan. Kemudian ke semua tayangan ini disambungkan untuk membentuk suatu sistem yang lengkap. Secara umumnya dokumentasi ini digunakan untuk bahagian bantuan pengguna dalam program. Bantuan pengguna adalah suatu kaedah sokongan pertolongan penggunaan program bagi mencapai salah satu atau kedua-dua objektif asasnya iaitu yang pertama menerangkan

konsep, kata utama dalam program, dan maksud cir-ciri dalam antara muka program dan yang kedua ialah menerangkan langkah-langkah penggunaan program seperti bagaimana menjalankan tugas-tugas tertentu serta langkah-langkah menyelesaikan masalah untuk penggunaan program.

6.3 Ringkasan

Daripada pelaksanaan sistem yang diterangkan, sistem ini dapat dilaksanakan dengan sempurna, hasil daripada menggunakan alatan pembangunan dan teknik aturcara yang dipilih. Ini dapat dilihat berdasarkan perisian ini mudah dilaksanakan dan tidak memerlukan perisian tambahan.

7. PENGUJIAN DAN PENYELENGGARAAN

7.1 Pengenalan

Pengujian merupakan satu proses untuk menguji keberkesanan sesuatu aturcara itu menjalankan fungsinya. Ia bertujuan untuk mencari ralat pada sesuatu sistem itu dan menjejaki kesilapan aturcara. Dengan ini ia dapat memastikan fungsi-fungsi yang dibina adalah bebas daripada sebarang masalah supaya sistem akan memberikan keputusan yang baik dan berkesan.

Proses pengujian merupakan elemen yang paling penting bagi memastikan sama ada sistem yang dihasilkan memenuhi kehendak pengguna atau tidak. Sistem yang berkualiti mampu menjalani apajua pengujian yang diberikan. Dengan itu, segala

spesifikasi, rekabentuk dan aturcara yang telah dilakukan sepanjang proses pembangunan sistem akan dapat diteliti semula.

Objektif utama dalam pengujian ini adalah untuk :

- i) Mengenalpasti ralat
Pemeriksaan secara teliti dilakukan ke atas setiap fungsi, perlakuan sistem dan mengenalpasti ralat yang ada.
- ii) Mengeluarkan ralat
Ralat dikeluarkan dengan cara debungging atau pengumpulan kod-kod selepas mencari sebab-sebab ralat.
- iii) Ujian regresi
Bagi melihat sama ada pembetulan pada ralat betul-betul menyelesaikannya atau memberi kesan sampingan pada bahagian kod yang lain.

7.2 Jenis-jenis Pengujian

7.2.1 Pengujian Unit

Langkah pertama di dalam proses pengujian adalah pengujian unit. Pengujian unit ini merangkumi pengujian ke atas setiap komponen fungsi aturcara itu sendiri dan diasingkan dengan fungsi-fungsi yang lain dalam aplikasi. Setiap fail dalam fungsi yang sama akan berinteraksi antara satu sama lain dan ia juga berinteraksi dengan fail pada fungsi yang lain.

7.2.2 Pengujian Modul

Setiap prosedur dan fungsi di dalam antaramuka sistem ini juga secara berasingan untuk memastikan ianya berfungsi dengan betul. Data-data ujian ini dimanipulasikan dengan menguji semula syarat-syarat dalam suatu segmen kod misalnya dalam segmen kawalan untuk memastikan segmen tersebut mengawal ralat dengan baik. Semua segmen kod yang telah diuji dengan baik menjamin kepantasan dan kebolehpercayaan sistem serta memudahkan ujian ke atas integrasi sistem.

7.2.3 Pengujian Integrasi

Pengujian integrasi merupakan pengujian terhadap satu sistem yang lengkap dimana komponen-komponen individu telah digabungkan dan dikombinasikan. Sistem ini dilihat sebagai satu hirarki komponen dimana setiap komponen dimiliki oleh satu lapisan rekabentuk. Dengan ini, ia dapat memberi satu gambaran yang sebenar apabila berlaku kegagalan sistem.

Terdapat empat pendekatan pada tahap ini.

- i) Integrasi Bawah-Atas (Bottom-Up Integration)
- ii) Integrasi Atas-Bawah (Top-Down Integration)
- iii) Integrasi Big-Bang
- iv) Integrasi Sandwich

Di dalam pengujian sistem ini, teknik integrasi Sandwich telah digunakan. Teknik ini merupakan suatu corak pengujian yang menggabungkan kaedah pengujian atas-bawah dan bawah-atas. Teknik ini dipilih kerana ia mempunyai banyak kelebihan berbanding dengan teknik-teknik yang lain, antaranya ialah ia membolehkan pengujian dilakukan pada peringkat yang

lebih awal dan komponen boleh diuji secara bersendirian ataupun bergabung. Selain daripada itu, ia dapat mengurangkan kesilapan dan menjadikan setiap fungsi itu lebih selamat dan aliran sistem akan menjadi lebih lancar.

7.2.4 Pengujian Sistem

Pengujian sistem bertujuan untuk memastikan bahawa sistem ini memenuhi keperluan pengguna. Terdapat dua jenis ujian pada peringkat ini iaitu pengujian fungsi dan pengujian pencapaian.

Pengujian fungsi adalah berdasarkan keperluan fungsi sistem dan ia lebih difokuskan fungsi-fungsi sesuatu aplikasi. Manakala pengujian pencapaian pula lebih tertumpu kepada keperluan yang bukan fungsi terhadap sesuatu aplikasi. Ia mengesan semua fungsi yang terhadap di dalam sistem berjalan dengan lancar disamping memastikan sistem mencapai objektifnya dan beroperasi dengan baik.

7.2.5 Pengujian Pengguna

Pengujian pengguna adalah merupakan pengujian penerimaan oleh pengguna terhadap sistem yang telah dibangunkan. Pengguna seharusnya dibiarkan secara bersendirian dalam melaksanakan ujian ini bagi memastikan ujian tersebut adalah tepat dan tidak 'bias'. Pada keseluruhannya, pengguna berpuas hati dan menunjukkan minat terhadap sistem yang dibangunkan walaupun pembangun merasakan masih terdapat banyak ruang yang perlu diperbaiki dan dipertingkatkan.

7.3 Perancangan Ujian

Perancangan ujian bertujuan untuk merekabentuk dan mengorganisasikan aktiviti-aktiviti ujian. Dengan ini proses pengujian dapat dijalankan dengan sempurna dan lancar. Langkah-langkah dalam perancangan ujian ada pada berikut :

- i) Membina objektif ujian
- ii) Merekabentuk kes ujian
- iii) Menulis kes ujian
- iv) Menguji kes ujian
- v) Melaksanakan ujian
- vi) Menilai keputusan ujian

7.4 Fasa Penyelenggaraan

Fasa ini melibatkan proses melakukan ubahsuai atau pembetulan ke atas ralat yang telah dikesan semasa proses pengujian. Proses ini dapat memastikan sistem atau aplikasi yang dibangunkan adalah mengikut spesifikasi dan mencapai objektif serta memuaskan hati pengguna. Proses penyelenggaraan ini hanya dilakukan apabila sistem tidak berjalan seperti yang diinginkan. Selepas proses ini selesai, dokumentasi dilakukan iaitu membuat manual pengguna yang digunakan sebagai panduan semasa menggunakan sistem.

7.5 Ringkasan

Bab ini menerangkan bagaimana pengujian dan penyelenggaraan dijalankan ke atas sistem yang telah dibangunkan. Fasa ini dijalankan bagi mengenalpasti sebarang ralat yang terdapat pada sistem ini bagi memastikan sistem ini mencapai objektifnya dan beroperasi dengan baik.

8. PENILAIAN SISTEM

8.1 Pengenalan

Proses penilaian sistem ini dilakukan sendiri oleh pembangun sistem bagi Bit Pointer 1 ini. Penilaian ini dilakukan untuk memberitahu kepada pengguna mengenai kekuatan sistem, had sistem dan juga peningkatan sistem pada masa hadapan.

8.2 Kekuatan Sistem

Sistem ini telah mencapai beberapa matlamat yang diingini. Ini merangkumi penggunaan elemen multimedia yang terdiri daripada grafik dan teks. Kekuatan sistem untuk aplikasi yang telah dibangunkan adalah seperti berikut :

8.2.1 Mesra Pengguna

Aplikasi yang dibangunkan ini amat mementingkan elemen multimedia yang boleh membantu dalam pendekatan mesra pengguna. Penggunaan grafik dan teks dapat membantu menarik tumpuan pengguna terhadap aplikasi ini. Selain itu, apabila tetikus dileretkan ke butang yang boleh diklik, tanda *cursor* secara automatiknya berubah ke bentuk jari.

8.2.2 Skrin yang Menarik

Aplikasi ini juga mempunyai skrin yang berwarna-warni untuk menarik minat pengguna. Secara tidak langsung, penggunaan rekabentuk skrin yang berlainan dapat mengelakkan pengguna merasa jemu semasa menggunakan aplikasi ini.

8.3 Had Sistem

Semua had yang dikatakan dibawah ini dapat diatasi jika tidak terdapat sebarang kekangan masa dan kemahiran.

8.3.1 Masalah Data

Aplikasi tidak memberi peluang kepada pengguna untuk mengawal secara terus nilai-nilai data, manakala hanya dengan manipulasi imej sahaja baru nilai data-data berubah.

8.3.2 Penggunaan Tetikus

Kebanyakan pelaksanaan modul di dalam aplikasi ini amat bergantung kepada tetikus dan ini mungkin akan menyukarkan golongan yang tidak mahir menggunakan tetikus.

8.4 Peningkatan Masa Hadapan

Aplikasi ini masih mempunyai beberapa kelemahan yang tertentu. Oleh kerana kekurangan masa dan kemahiran, banyak spesifikasi sistem tidak dapat dilaksanakan dengan sepenuhnya. Beberapa rancangan akan dikenalpasti untuk diimplementasikan pada masa yang akan datang.

8.5 Ringkasan

Bab ini memerangkan kelebihan dan had yang terdapat dalam sistem ini supaya dapat diperbaharui pada masa akan datang. Beberapa rancangan juga akan dikenalpasti supaya sebuah aplikasi yang berkualiti dapat dihasilkan pada masa hadapan.

9. MASALAH, PENYELESAIAN DAN KESIMPULAN

9.1 Masalah-masalah

Di dalam usaha pembangun sistem untuk membangunkan aplikasi ini, pelbagai dugaan dan rintangan telah dihadapi. Namun masalah-masalah tersebut telah diselesaikan satu persatu dan ianya telah mengambil masa yang agak lama sekiranya masalah yang dihadapi itu adalah bersifat kritikal. Berikut merupakan antara masalah-masalah yang dihadapi dan cara penyelesaiannya :

9.1.1 Pengetahuan

Pengetahuan pembangun sistem dalam perisian yang digunakan tidak begitu meluas. Justeru itu, pembangun sistem memerlukan daya usaha sendiri untuk mempelajarinya. Perisian pembangunan seperti Microsoft Visual C++ adalah agak sukar untuk dipelajari tetapi dengan sokongan tutorial dan contoh program dapat membantu memahami konsep-konsep dengan mudah. Walaubagaimanapun, ianya memerlukan masa yang agak lama untuk memahirkan diri bagi menggunakannya.

Penyelesaian :

Masalah tersebut telah ditangani dengan merujuk kepada buku panduan Microsoft Visual C++ di mana, di dalam buku tersebut turut diterangkan berkenaan “library” yang terletak dengan kelas C++ iaitu Microsoft Foundation Class (MFC) Library. Dengan itu pembangun perlu memahami dan mencubanya sendiri bagi menambahkan kemahiran dalam menggunakan perisian tersebut. Di samping itu, pembangun sistem turut mendapatkan bantuan daripada rakan-rakan yang mahir menggunakan perisian tersebut.

9.1.2 Paparan Maklumat

Maklumat yang dipaparkan kadang kala banyak sehingga melebihi tetingkap aplikasi walaupun telah disusun rapi untuk menjimatkan ruang paparan. Begitu juga dengan imej yang dipaparkan yang berbagai jenis saiz resolusinya yang juga ada melebihi saiz tetingkap aplikasi. Bagi paparan imej, adalah amat penting untuk mengekalkan ketepatan nilai resolusi imej kerana dari imej akan dimanipulasi kepada bentuk paparan yang baru. Maka amat penting imej tidak boleh diubah saiz resolusi atau dimampatkan resolusi jika hendak dimuatkan kepada saiz tetingkap aplikasi yang sedia ada. Data-data maklumat juga berjenis berubah-ubah jumlahnya. Ini akan menimbulkan kesulitan jika saiz tetingkap aplikasi tidak cukup untuk memaparkan kesemua data tersebut.

Penyelesaian :

Masalah ini dapat diselesaikan dengan menggunakan teknik gulungan (*scroll*) pada paparan imej dan maklumat. Jika paparan imej dan juga paparan maklumat melebihi saiz tetingkap, format gulungan akan digunakan pada paparan tersebut dan menampung keseluruhan saiz imej dan juga saiz maklumat. Cara menggunakan paparan gulungan amat mudah iaitu pengguna hanya perlu menarik bar gulungan (*scroll bar*) menegak ke atas atau ke bawah untuk melihat paparan dari atas ke bawah dan menarik bar gulungan melintang ke kanan atau ke kiri untuk melihat paparan dari kiri ke kanan. Kaedah gulungan ini adalah bersifat dinamik kerana ia bergantung kepada saiz dimensi objek yang ingin dipaparkan. Ia akan hanya digunakan apabila saiz itu menepati syarat yang ditetapkan sahaja.

9.2 Kesimpulan

Secara keseluruhannya, selaras dengan keadaan negara kita yang sedang menuju ke era teknologi maklumat, maka secara langsung atau tidak kita sebagai rakyat Malaysia harus memainkan peranan yang penting untuk membawa negara kita ke matlamat yang diimpikan. Pembangunan Bit Pointer 1 ini merupakan salah satu usaha bagi mempertingkatkan kemahiran dalam penggunaan teknologi terkini di kalangan masyarakat.

Peluang untuk mengaplikasi pengetahuan yang telah dipelajari sepanjang pengajian dalam bidang sains komputer seperti Analisis dan Rekabentuk Sistem, Kejuruteraan Perisian dan Pengaturcaraan Berorientasikan Objek telah digunakan dalam projek ini.

Pembangun sistem juga berpeluang menggunakan perisian hebat yang semakin popular pada masa kini iaitu Microsoft Visual C++. Ianya mungkin berguna pada masa akan datang terutamanya di alam pekerjaan.

Pengalaman melaksanakan sebuah sistem secara individu amat berguna sekali kerana ianya telah memupuk sifat keyakinan diri yang mendalam dan kekangan masa.

SENARAI RUJUKAN

M. A. Sid-Ahmed (1995). *Image Processing – Theory, Algorithms, & Architectures*.

Stanley Thornes Publishers.

Holzner. S.(1994). *Heavy Metal Visual C++ Programming*. IDG Books Worldwide Inc.

Genie Form Processing. <http://www.stiscan.com/forms/genieinf.html>

The Code Project. <http://www.codeproject.com>

Code Guru. <http://www.codeguru.com>

Canny Edge Detector. <http://black.csl.uiuc.edu/~yima/imgproc/index.html>

LAMPIRAN A

Contoh Kod Program Pada Sistem Imej Kepada Data

Kod aturcara untuk memaparkan paparan imej :

```
void CBitPointerIView::OnDraw(CDC* pDC)
{
    CBitPointerIDoc* pDoc = GetDocument();
    ASSERT_VALID(pDoc);
    // TODO: add draw code for native data here

    CRect rect;
    GetClientRect(&rect);

    CDC dc;
    CBitmap bmp, *oldbmp;
    dc.CreateCompatibleDC(pDC);
    bmp.CreateCompatibleBitmap(pDC, rect.Width(), rect.Height());
    oldbmp = dc.SelectObject(&bmp);

    dc.FillSolidRect(rect, RGB(255,255,255));

    int cxDIB = 0, cyDIB = 0;
    HDIB hDIB = pDoc->GetHDIB();
    if (hDIB) {
        LPSTR lpDIB = (LPSTR) ::GlobalLock((HGLOBAL) hDIB);
        cxDIB = (int) ::DIBWidth(lpDIB);
        cyDIB = (int) ::DIBHeight(lpDIB);
        ::GlobalUnlock((HGLOBAL) hDIB);
        CRect rcDIB;
        rcDIB.top = rcDIB.left = 0;
        rcDIB.right = cxDIB;
        rcDIB.bottom = cyDIB;
        CRect rcDest = rcDIB;
        if (cxDIB > rect.Width() || cyDIB > rect.Height())
        {
            rect.right = cxDIB;
            rect.bottom = cyDIB;
            bmp.CreateCompatibleBitmap(pDC, cxDIB, cyDIB);
            oldbmp = dc.SelectObject(&bmp);
        }
        if ((cxDIB * 2 <= rect.Width()) || !(HBITMAP(m_bmpEdges))) {
            ::PaintDIB(dc.m_hDC, &rcDest, hDIB,
                &rcDIB, pDoc->GetDocPalette());
        }
    }

    if ((HBITMAP)m_bmpEdges) {
        CDC cdc;
        CBitmap *oldbmp;

        cdc.CreateCompatibleDC(pDC);
        oldbmp = cdc.SelectObject(&m_bmpEdges);

        if (cxDIB * 2 < rect.Width()) {
            dc.BitBlt(cxDIB+1,0,cxDIB,cyDIB+m_iOffset,&cdc,0,0,SRCCOPY);
        } else {
            bmp.CreateCompatibleBitmap(pDC, cxDIB, cyDIB);
            oldbmp = dc.SelectObject(&bmp);
            dc.BitBlt(0,0,cxDIB,cyDIB+m_iOffset,&cdc,0,0,SRCCOPY);
        }
    }

    pDC->BitBlt(0,0,rect.Width(),rect.Height(),&dc,0,0,SRCCOPY);
    dc.SelectObject(oldbmp);
}
```



```
SetScrollSizes(MM_TEXT, GetDIBSizeTotal(cxDIB, cyDIB));
```

Kod aturcara untuk memaparkan paparan data:

```
void CHexviewView::OnDraw(CDC* pDC)
{
    CBitPointer1Doc* pDoc = GetDocument();
    ASSERT_VALID(pDoc);

    DispData(pDC, m_iFirstLine, pDoc->TotalFileLength(), m_cfFixedFont,
                                                    m_cfBoldFont, m_iWndLines,
m_iFontHeight, m_iCurrentOffset, m_iFontWidth);
}

void CHexviewView::DispData(CDC* pDC, int iFirstLine, int iTotalLines,
                                                    CFont *cfFixedFont, CFont *cfBoldFont,
int iWndLines, int iFontHeight, int
iCurrentOffset,
                                                    int iFontWidth, int iLine)
{
    CFont *pOldFont = pDC->SelectObject(cfFixedFont);

    CRect rcWnd;
    GetClientRect(&rcWnd);

    int iOffset = iFirstLine * 16 + iCurrentOffset;
    int iRemaining = iTotalLines - iOffset;
    CBitPointer1Doc* pDoc = GetDocument();
    BYTE *pData = pDoc->AdjustPointerAbsolute(0);
    bool bIgnoreThisChar = false;

    int x=0;
    if (!pData)
    {
    }
    else
    {
        char buf[100], buf2[100];
        while ((iRemaining > 0) && (iLine < iWndLines))
        {
            sprintf(buf, "%8.8lX ", iOffset);
            BYTE *p = pData + iOffset;
            int iCount = iRemaining;
            for (int i = 0; i < 16; i++)
            {
                if (iCount > 0)
                {
                    sprintf(&buf[strlen(buf)],
                        "%2.2X ",
                        *p);
                }
                else
                {
                    strcat(buf, " ");
                }
            }
        }
    }
}
```

```

    }
    iCount--;
    p++;
}
strcat(buf, " ");
p = pData + iOffset;
iCount = iRemaining;
for (i = 0; i < 16; i++)
{
    if ((bDBCS) && (IsDBCSLeadByte(*p)) && !bIgnoreThisChar)
    {
        sprintf(&buf[strlen(buf)], "%c%c", *p, *(p + 1));
        iCount-=2;
        p+=2;
        i++;
        if (i >= 16)
            bIgnoreThisChar = true;

        continue;
    }
    if ((iCount > 0) && isprint(*p) && !bIgnoreThisChar)
    {
        sprintf(&buf[strlen(buf)], "%c", *p);
    }
    else if (iCount > 0)
    {
        strcat(buf, ".");
    }
    else
    {
        strcat(buf, " ");
    }
    bIgnoreThisChar = false;
    iCount--;
    p++;
}

#define LINE_LENGTH 78
if ((i = strlen(buf)) < LINE_LENGTH)
{
    memset(&buf[i], ' ', LINE_LENGTH-i);
    buf[LINE_LENGTH]=0;
}

if ((iOffset + 16 >= pDoc->m_lStartOffset) && (iOffset < pDoc->m_lEndOffset))
{
    strcpy(buf2, buf);
    CSize szExtent;
    int nStartOnLine = max(0, pDoc->m_lStartOffset - iOffset);
    int nItemsOnLine = min(pDoc->m_lEndOffset, iOffset + 16) -
max(pDoc->m_lStartOffset, iOffset);

    pDC->TextOut(0, iLine * iFontHeight, buf, nStartOnLine * 3 + 10);
    szExtent = pDC->GetTextExtent(buf, nStartOnLine * 3 + 10);
    memcpy(buf, buf2 + nStartOnLine * 3 + 10, (16 - nStartOnLine) * 3 + 2);
    buf[(16 - nStartOnLine) * 3 + 2] = 0;
    pDC->SelectObject(cfBoldFont);
    pDC->TextOut(szExtent.cx,
iLine * iFontHeight, buf, nItemsOnLine * 3);
    szExtent += pDC->GetTextExtent(buf, nItemsOnLine * 3);
    pDC->SelectObject(cfFixedFont);
    strcpy(buf, buf2 + (nStartOnLine + nItemsOnLine) * 3 + 10);

    pDC->TextOut(szExtent.cx, iLine * iFontHeight, buf, strlen(buf));
    pDC->MoveTo(0, (iLine + 1) * iFontHeight);
}

```

```

        else
            pDC->TextOut(0, iLine * iFontHeight, buf, strlen(buf));

        iOffset += 16;
        iRemaining -= 16;
        iLine++;
    }
    if (iLine < iWndLines)
    {
        CRect rc = rcWnd;
        rc.top = iLine * iFontHeight;
        pDC->ExtTextOut(0, iLine * iFontHeight, ETO_OPAQUE,
            &rc, "", 0, NULL);
    }
}
pDC->SelectObject(pOldFont);
}

```

```

BYTE *CBitPointerIDoc::AdjustPointerAbsolute(int iPosition)
{
    m_iCurrentPointer=iPosition;
    return &m_lplmage[m_iCurrentPointer];
}

```

Kod aturcara untuk fungsi edge detection:

```

void CBitPointerIView::OnEditEgdedetect()
{
    // TODO: Add your command handler code here

    CWaitCursor wait;

    CMainFrame* pMfm = STATIC_DOWNCAST(CMainFrame, AfxGetMainWnd());

    m_iOffset = 0;
    int thresh = pMfm->m_iThreshold;

    CDC sc1,sc2;
    CClientDC dc(this);

    sc1.CreateCompatibleDC(&dc);
    sc2.CreateCompatibleDC(&dc);

    CBitmap *srcIold, bmsrc, *src2old;

    int cxDIB, cyDIB;

    CBitPointerIDoc* pDoc = GetDocument();
    HDIB hDIB = pDoc->GetHDIB();

    GetDIBSize(cxDIB, cyDIB);

    if ((HBITMAP)(m_bmpEdges)) m_bmpEdges.Detach();

    bmsrc.CreateCompatibleBitmap(&dc, cxDIB, cyDIB);
}

```

```

m_bmpEdges.CreateCompatibleBitmap(&dc, cxDIB, cyDIB);

src1old = sc1.SelectObject(&bmsrc);
src2old = sc2.SelectObject(&m_bmpEdges);

::PaintDIB(sc1.m_hDC, &CRect(0,0,cxDIB,cyDIB), hDIB,
&CRect(0,0,cxDIB,cyDIB), pDoc->GetDocPalette());

float red, blue, green;
//      int red, blue, green;
for (int i=1;i<cxDIB-1;i++) {
    for (int j=1;j<cyDIB-1;j++) {
        red = blue = green = 0;

        UpdateTotals(red, blue, green, m_fFilter[4], sc1.GetPixel(i,j));

        UpdateTotals(red, blue, green, m_fFilter[0], sc1.GetPixel(i-1,j));
        UpdateTotals(red, blue, green, m_fFilter[2], sc1.GetPixel(i+1,j));
        UpdateTotals(red, blue, green, m_fFilter[6], sc1.GetPixel(i,j-1));
        UpdateTotals(red, blue, green, m_fFilter[8], sc1.GetPixel(i,j+1));

        UpdateTotals(red, blue, green, m_fFilter[1], sc1.GetPixel(i-1,j-1));
        UpdateTotals(red, blue, green, m_fFilter[3], sc1.GetPixel(i+1,j-1));
        UpdateTotals(red, blue, green, m_fFilter[5], sc1.GetPixel(i-1,j+1));
        UpdateTotals(red, blue, green, m_fFilter[7], sc1.GetPixel(i+1,j+1));

        // Get the threshold value, and check whether any of the
        // values exceed the threshold value.
        if (Plot(int(red),int(green),int(blue),thresh)) {
            if (m_bColour)
                sc2.SetPixel(i,j, RGB(int(red),int(blue),int(green)));
            else sc2.SetPixel(i,j, m_crPlot);
        } else {
            sc2.SetPixel(i,j, m_crNoPlot);
        }
    }

    pMfm->PercentComplete(i / float(cxDIB) * 100);
}

CBrush br(RGB(0,0,0));
sc2.FrameRect(&CRect(0,0,cxDIB, cyDIB), &br);

sc1.SelectObject(src1old);
sc2.SelectObject(src2old);

Invalidate();
pDoc->UpdateAllViews(NULL);

pMfm->SetStatusMessage(AFX_IDS_IDLEMESSAGE);

}

void CBitPointerView::UpdateTotals(float &r, float &g, float &b, float filter, COLORREF cr) {
    r += GetRValue(cr)*filter;
    b += GetBValue(cr)*filter;
    g += GetGValue(cr)*filter;
}

bool CBitPointerView::Plot(int r, int g, int b, int t) {
    return (r > t && g > t && b > t);
}

```


Kod aturcara untuk membuka fail:

```
BOOL CBitPointerIDoc::OnOpenDocument(LPCTSTR lpszPathName)
{
    if (!CDocument::OnOpenDocument(lpszPathName))
        return FALSE;

    m_hDIB = NULL;
    m_palDIB = NULL;

    DeleteContents();
    UpdateAllViews(NULL);

    CFile file;
    if (!file.Open(lpszPathName, CFile::modeRead)) {
        AfxMessageBox("Problem opening file.");
        return(FALSE);
    }

    m_hDIB = ::ReadDIBFile(file);
    file.Close();

    InitDIBData();
    if (!m_hDIB) {
        AfxMessageBox("Error reading Bitmap section.");
        return(FALSE);
    }

    m_csFileName = (CString *)0;
    if
(INVALID_HANDLE_VALUE==(m_hFile=CreateFile(lpszPathName,GENERIC_READ,FILE_SHARE_READ|FILE_SHARE_WRITE,NULL,
        OPEN_EXISTING,FILE_ATTRIBUTE_NORMAL,NULL)))
    {
        AfxMessageBox(IDS_ERR_OPEN);
        return(FALSE);
    }
    m_hFileMapping = CreateFileMapping(m_hFile,NULL,PAGE_READONLY,0,0,NULL);
    if (!m_hFileMapping)
    {
        AfxMessageBox(IDS_ERR_MAPPING);
        CloseHandle(m_hFile);
        return(FALSE);
    }
    m_lpImage = (BYTE *)MapViewOfFile(m_hFileMapping,FILE_MAP_READ,0,0,0);
    if (!m_lpImage)
    {
        CloseHandle(m_hFileMapping);
        CloseHandle(m_hFile);
        AfxMessageBox(IDS_ERR_MAPVIEW);
        return(FALSE);
    }

    AfxGetApp()->AddToRecentFileList(lpszPathName);
    m_csFileName = new CString(lpszPathName);
    SetPathName(lpszPathName);
    m_iCurrentPointer = 0;
    DWORD dwFileSizeHigh;
    m_iFileSize = GetFileSize(m_hFile,&dwFileSizeHigh);
    m_iBlockSize = min(HEXVIEW_BLOCK_SIZE, m_iFileSize);

    return TRUE;
}
```

LAMPIRAN B

Senarai Kandungan Manual Pengguna

Tajuk

Halaman

1.0	Pendahuluan	3
2.0	Keperluan Sistem	3
3.0	Kaedah Pemasangan Sistem	4
4.0	Permulaan Sistem dan Menu Utama	5
4.1	Skrin Menu Utama	5
4.2	Skrin Paparan Imej dan Data	6
4.3	Skrin Paparan Imej Edge Detect	8
4.4	Skrin Cetakan Imej	9
4.5	Skrin Cetakan Data	10

1.0 Pendahuluan

Sistem ini dibangun dengan tujuan untuk dapatkan data objek di dalam imej grafik. Imej akan dikecam untuk mengesan sebarang objek yang wujud dan kemudian lokasi objek dicatatkan. Data pada lokasi itu akan diambil sebagai data objek.

Manual pengguna ini disediakan adalah untuk membekalkan maklumat-maklumat yang perlu diketahui oleh pengguna sebelum menggunakan sistem ini. Ia adalah sebagai garis panduan untuk memperolehi serba sedikit maklumat mengenai perkakasan, perisian dan sistem pengendalian yang perlu dengan merujuk kepada manual pengguna ini.

Disamping itu, penerangan mengenai teknik pemasangan dan pelaksanaan sistem turut disediakan untuk membantu melancarkan kerja-kerja pengguna semasa melaksanakan sistem ini.

2.0 Keperluan Sistem

Keperluan sistem yang dicadangkan ini adalah keperluan sistem yang membolehkan sistem Bit Pointer1 ini dijalankan pada tahap yang optimum. Ini adalah penting supaya sistem ini dapat memberi dan menunjukkan prestasinya yang sebenar.

Keperluan sistem adalah seperti berikut :

- Pemproses komputer Intel Pentium I / II / Celeron / III / IV
- 32-128 MB memori
- Monitor
- Paparan warna 16 bit
- 1 MB cakera keras

- Pemacu cakera padat 20X
- Papan kekunci
- Tetikus
- Windows 95 / 98 / ME / 2000

3.0 Kaedah Pemasangan Sistem

Berikut adalah langkah-langkah pemasangan sistem :

- 1) Masukkan cakera padat ke dalam pemacu cakera padat
- 2) Pemasangan automatik akan dijalankan sejourus pemacu cakera padat membaca cakera padat. Jika pemasangan automatik tidak berlaku, pemasangan secara manual boleh dilakukan dengan membuka kandungan cakera keras dan klik dua kali pada fail Setup.exe yang berada di dalam direktori :

D:\Setup.exe (dimana D: ialah nama pemacu bagi pemacu cakera padat)

- 3) Ikuti arahan pemasangan sehingga akhir prosesnya.
- 4) Setelah selesai proses pemasangan ke dalam komputer, maka sistem telah sedia untuk digunakan.

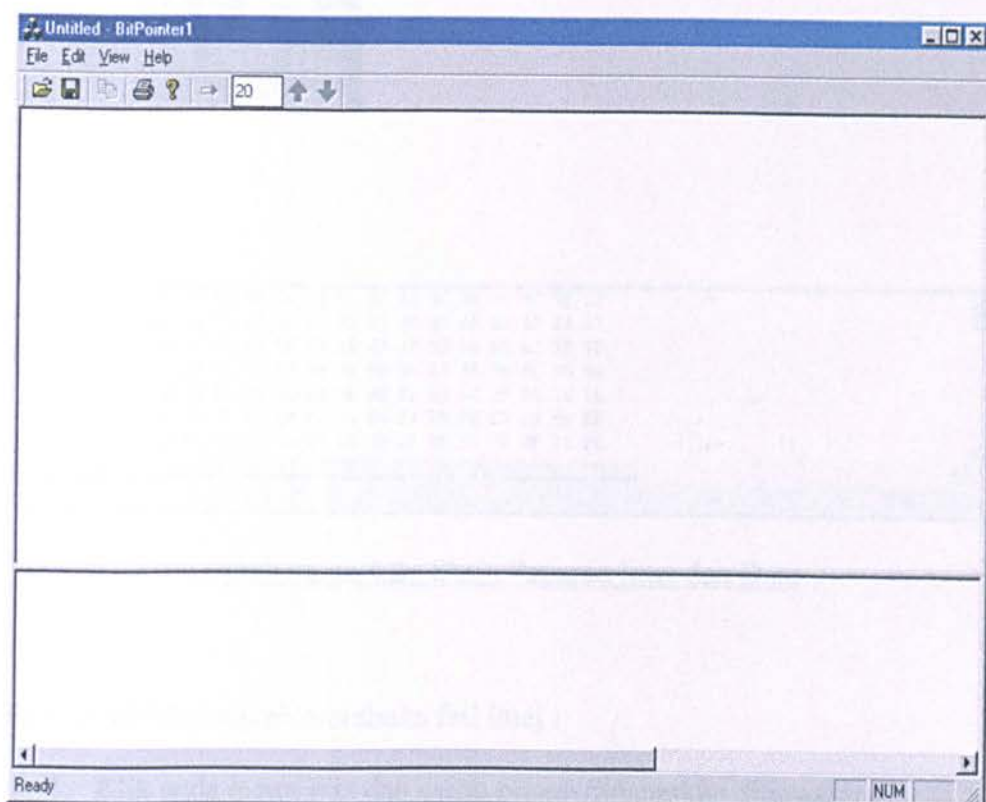
Berikut adalah tiga cara pilihan untuk membuka program sistem :

- 1) Klik dua kali pada ikon program sistem pada desktop dan sistem akan mula dijalankan.

- 2) Klik pada butang “Start” dan pergi ke direktori program. Klik pada ikon program sistem dan sistem akan mula dijalankan.
- 3) Buka “My Computer” dan pergi ke direktori program sistem. Klik dua kali pada ikon program sistem dan sistem akan mula dijalankan.

4.0 Permulaan Sistem dan Menu Utama

4.1 Skrin Menu Utama

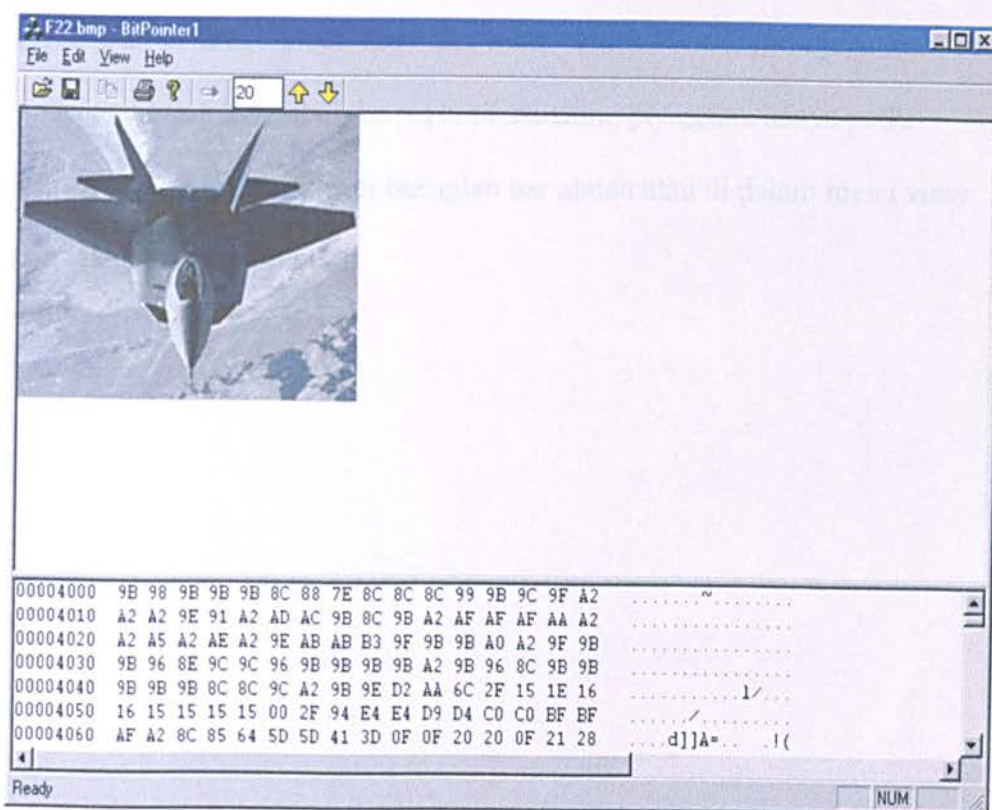


Gambarajah 4.1 : Paparan Skrin Menu Utama

Pada skrin utama ini, terdapat dua bahagian paparan sistem secara susunan melintang iaitu paparan pertama adalah untuk memaparkan imej grafik dan paparan kedua adalah untuk memaparkan maklumat data dalam bentuk teks. Menu utama disediakan seperti file, edit, view, dan help untuk interaktif sistem dengan pengguna. Butang-butang ikon dalam bar alat menyokong

sesetengah fungsi dalam menu utama bagi hubungan terus dengan fungsi sistem.

4.2 Skrin Paparan Imej dan Data



Gambarajah 4.2 : Skrin Paparan Imej dan Data

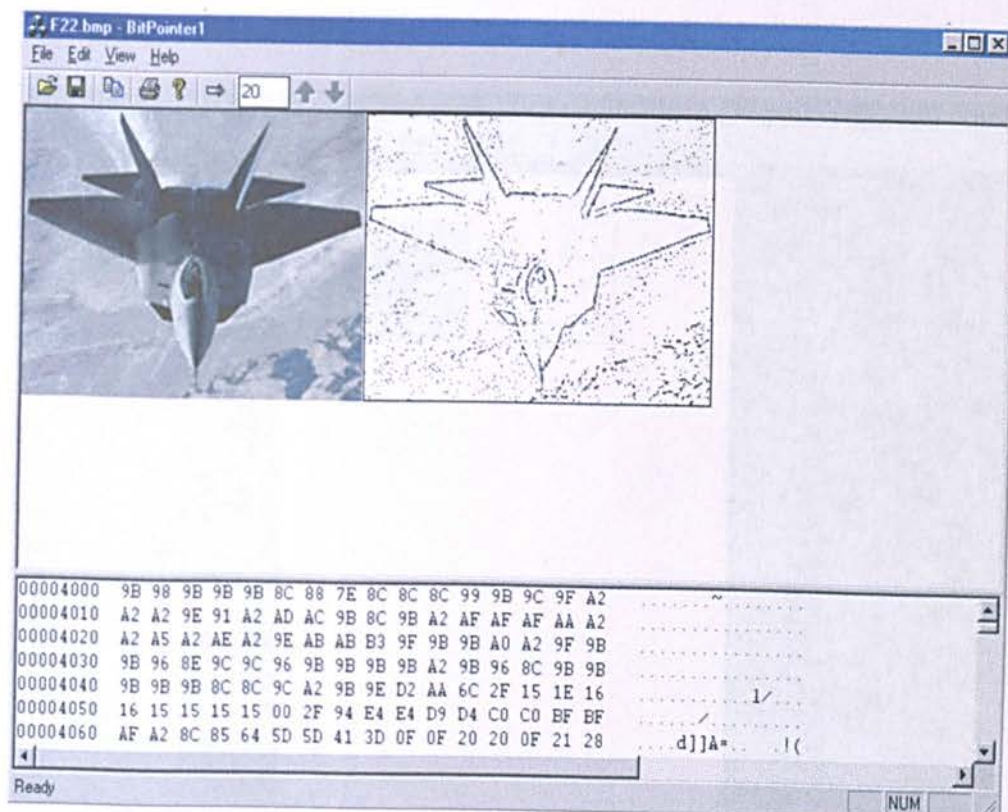
Berikut adalah langkah membuka fail imej :

1. Klik pada menu File dan menu popup dipaparkan dibawahnya.
2. Klik pada Open dan pilih fail imej yang ingin dipaparkan.
3. Klik OK dan imej dipaparkan dan juga datanya seperti contoh paparan di gambarajah 4.2.

Data yang dipaparkan boleh dilihat sehingga data ke 16,382 kerana paparan data telah ditetapkan nilai maksimum paparan semasa. Ini adalah untuk

memudahkan pengguna melihat data yang terlalu banyak di bahagikan secara berjujukan. Paparan data disetkan maksimum 16,382 bagi setiap jujukan sehingga data yang terakhir. Oleh itu, pengguna jika hendak melihat paparan yang seterusnya, pengguna hanya perlu menekan butang Next di bahagian bar alatan atau di dalam menu view dan tekan Next Block. Begitu juga jika pengguna hendak kembali ke paparan sebelum, pengguna hanya perlu menekan butang Previous di bahagian bar alatan atau di dalam menu view dan tekan Previous Block.

4.3 Skrin Paparan Imej Edges Detect

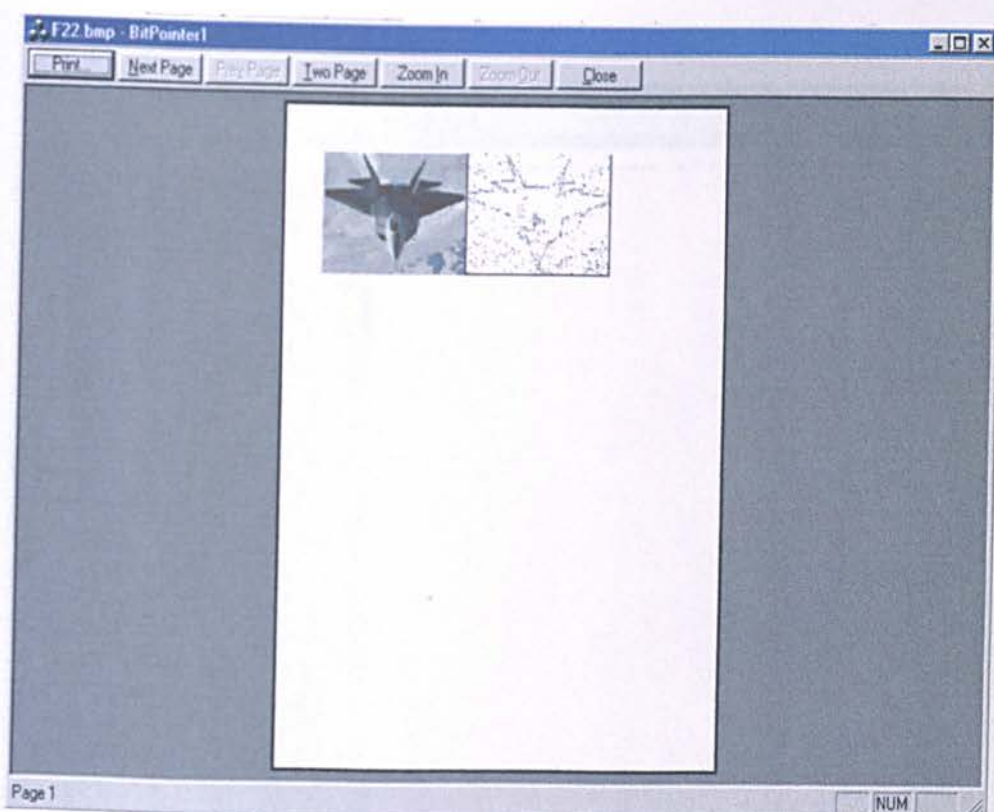


Gambarajah 4.3 : Skrin Paparan Imej Edge Detect

Berikut adalah langkah-langkah melakukan edge detect :

1. Dengan imej telah dipaparkan, tekan butang Edge Detect di bahagian bar alatan atau dalam menu Edit dan klik Edge Detect.
 2. Penunjuk peratusan proses edge detection ditunjukkan dibahagian bar status dari mula proses sehingga proses lengkap dilakukan.
 3. Imej baru akan dipaparkan di paparan imej menggantikan imej asal.
- Jika ruang paparan mencukupi untuk memaparkan imej asal dan imej baru, maka imej baru akan dipaparkan bersebelahan dengan imej asal.

4.4 Skrin Cetakkan Imej

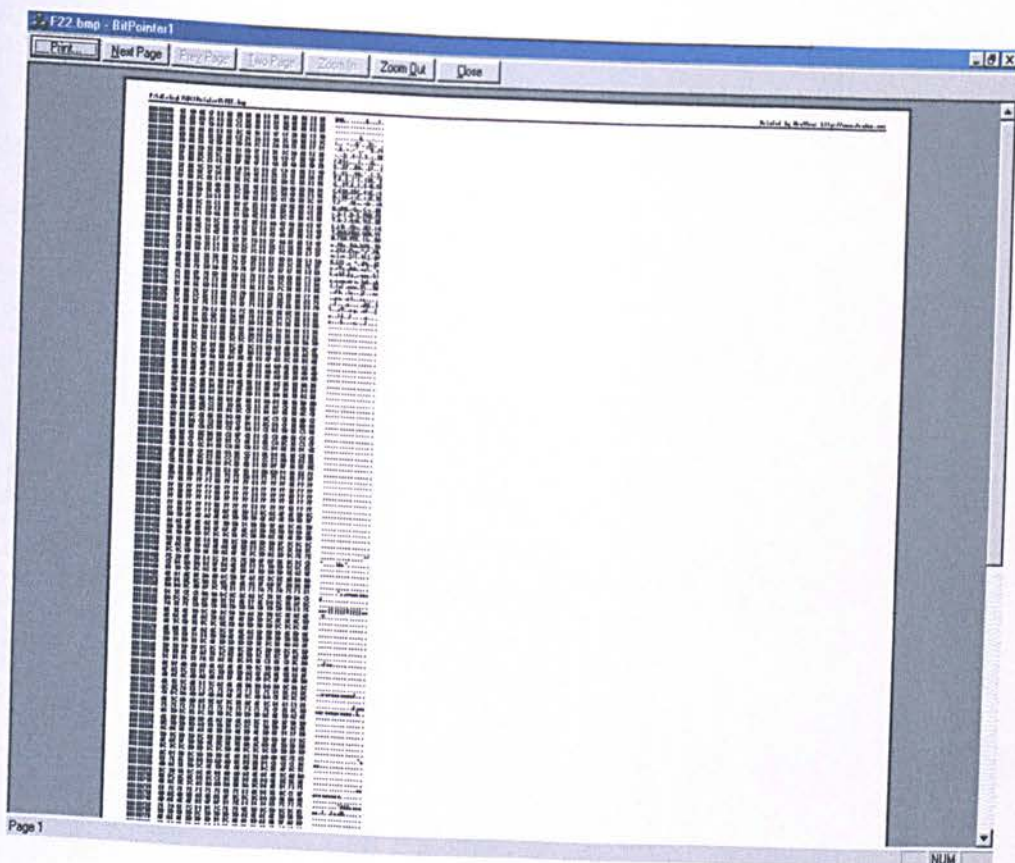


Gambarajah 4.4 : Skrin Cetakan Imej

Berikut adalah langkah untuk melakukan cetakan imej :

1. Klik pada mana-mana bahagian dalam paparan imej. Ini adalah penting supaya program fokus pada paparan imej.
2. Klik butang Print pada bar alatan atau dalam menu File dan klik pada Print.
3. Satu skrin print akan dipaparkan. Lakukan ubahsuai (jika perlu) mengikut kehendak pengguna. Klik OK dan cetakan akan diproses.

4.5 Skrin Cetakan Data



Gambarajah 4.5 : Skrin Cetakan Data

Berikut adalah langkah-langkah melakukan cetakan data :

1. Klik pada mana-mana bahagian dalam paparan data. Ini adalah penting supaya program fokus pada paparan data.
2. Klik butang Print pada bar alatan atau dalam menu File dan klik pada Print.
3. Satu skrin print akan dipaparkan. Lakukan ubahsuai (jika perlu) mengikut kehendak pengguna. Klik OK dan cetakan akan diproses.